

**DEDICATED PROCESSOR FOR EFFICIENT PROCESSING OF DOCUMENTS  
ENCODED IN A MARKUP LANGUAGE**

Publication number: WO02091170

Publication date: 2002-11-14

Inventor: JAMES ZACKARY ANTONE; RAJARAMAN BALA

Applicant: IBM (US); IBM UK (GB)

Classification:



- international: G06F17/22; G06F17/27; G06F17/22; G06F17/27;  
(IPC1-7): G06F9/44; G06F17/30

- european: G06F17/22M; G06F17/22T; G06F17/27A; G06F17/27A6

Application number: WO2002GB01978 20020501

Priority number(s): US20010848828 20010504

Also published as:

 US7013424 (B2)  
 US2004205694 (A)

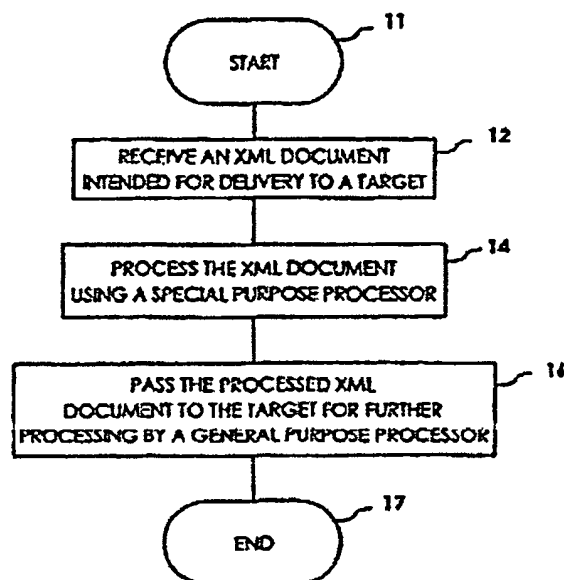
Cited documents:

 WO0052564  
 GB2357348

Report a data error he

**Abstract of WO02091170**

A dedicated processor for efficient processing of documents encoded in a markup language, such as XML. The dedicated processor is capable of performing traditional parsing, transformation and manipulation processes on the document. The special purpose processor frees a general purpose processor to perform other tasks, resulting in an increase in system performance. In one embodiment, the dedicated processor includes a general purpose processor and suitable software which is provided in addition to the general purpose processor which has been traditionally used for processing. In such an embodiment, the dedicated processor may be implemented in a multi-processor system. In another embodiment, the dedicated processor is implemented in special purpose hardware, e.g. as an integrated circuit embodied in silicon in one or more chips. In either embodiment, the dedicated processor may be provided to offload processing locally or remotely.



Data supplied from the esp@cenet database - Worldwide

→ abstract  
for KR 2003-0094320

특2003-0094320

(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(51) Int. Cl.  
G06F 17/00

(11) 공개번호 특2003-0094320  
(43) 공개일자 2003년12월11일

(21) 출원번호 10-2003-7012977  
(22) 출원일자 2003년10월02일  
    변역문제출일자 2003년10월02일  
(86) 국제출원번호 PCT/GB2002/01978 (87) 국제공개번호 WO 2002/91170  
(86) 국제출원출원일자 2002년05월01일 (87) 국제공개일자 2002년11월14일

(30) 우선권주장 09/848,828 2001년05월04일 미국(US)  
(71) 출원인 인터내셔널 비지네스 머신즈 코퍼레이션  
    미국 10504 뉴욕주 아몬크  
(72) 발명자 제임스잭캐리안톤  
    미국27616노스캐롤라이나주롤리외일드우드포레스트로드6608  
    라자리안발라  
    미국27513노스캐롤라이나주캐리크롬웰코트109  
(74) 대리인 김창세, 장성규, 김원준

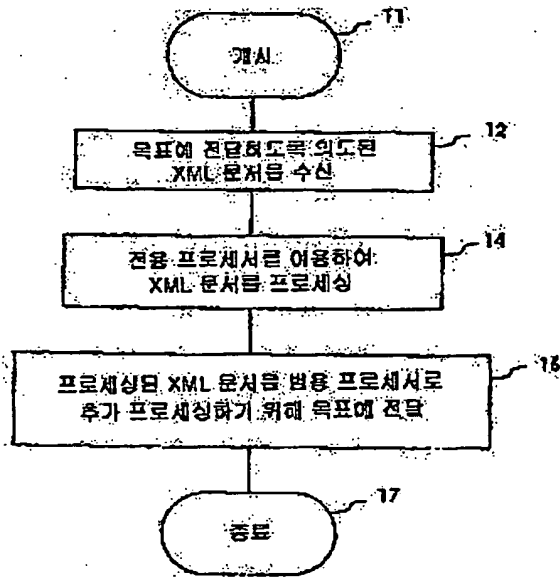
심사원구 : 없음

(54) 마크업 언어로 인코딩된 문서의 효율적인 프로세싱 방법 및 그 시스템

요약

XML 등의 마크업 언어(markup language)로 인코딩된 문서를 효율적으로 프로세싱하는 전용 프로세서(dedicated processor)를 제공한다. 이 전용 프로세서는 문서에 대해서 통상적인 파싱(parsing), 변환 및 조작 프로세스를 수행할 수 있다. 전용 프로세서는 범용 프로세서가 다른 작업을 자유롭게 수행하게 하는 것에 의해서 시스템 성능을 향상시킨다. 일 실시예에서, 전용 프로세서는 범용 프로세서(general purpose processor)와 물리적으로 프로세싱에 이용되어 온 범용 프로세서에 추가하여 제공되는 적합한 소프트웨어를 포함한다. 이러한 실시예에서, 전용 프로세서는 다중 프로세서 시스템으로 구현될 수 있다. 다른 실시예에서, 전용 프로세서는 예를 들면, 하나 이상의 컴퓨터 칩 내부에 매립된 집적 회로 등으로서 전용 하드웨어로 구현될 수 있다. 어느 실시예에서도, 전용 프로세서는, 국부적으로 또는 원격으로 프로세싱을 오프로딩(offload)하는 데 제공될 수 있다.

0000



명세서

기술분야

본 발명은, 일반적으로 XML(Extensible Markup Language) 등의 마크업 언어로 인코딩된 문서에 관한 것이고, 세부적으로는, 통신 네트워크 등의 XML 환경에서 XML 문서를 프로세싱하는 것에 관한 것이다.

배경기술

예를 들면, 네트워크 컴퓨터 등의 비즈니스용 및 소비자용 분산형 컴퓨터는 최근 대단한 인기를 얻고 있다. 비즈니스용에 있어서는, 분산형 컴퓨터의 컴퓨터 소자 사이의 네트워크 상호 작용에는 2개의 주요 카테고리, 다시 말해, 사용자를 비즈니스 프로세스에 접속시키는 것과 비즈니스 프로세스 소자를 상호 접속시키는 것이 존재한다. 전자의 예는, 사용자가 웹 브라우저 소프트웨어를 이용하여, HTTP(HyperText Transport Protocol)에 의해서 전송된 HTML(HyperText Markup Language) 데이터 포맷을 이용하는 웹 브라우저에서 비즈니스 데이터 및 애플리케이션과 상호 작용할 수 있게 하는 통상적인 웹(Web)이다. 후자의 예는, 표준화된 전자 포맷(ANSI X.12 또는 EDIFACT 등)에서 존재하는 청구서, 구매 주문서, 송장(invoice), 출하 통지(shipping notification) 등의 문서를, X.400, SNADS, THR, SMTP 등의 프로토콜에 의해 조직적 프로세스 사이에서 이동시키는 통상적인 전자 문서 상호 교환(Electronic Document Interchange: EDI)이다. 네트워크 상호 작용의 카테고리의 양자에 있어서는, HTTP 웹 전송 프로토콜 및 XML(Extensible Markup Language)로 알려진 공통 데이터 포맷을 이용하는 경향으로 기울어지고 있다.

XML은 태그 언어(tag language)로서, 정보를 범위 지정(delimit)(또는 '마크업(mark up)')하기 위해서, 태그(tag)로 지정되는 특수 지정 구성(specially-designated constructs)을 이용하는 언어이다. 일반적인 경우에, 태그는, 태그와 연관된 데이터를 식별하는 키워드이고, 전형적으로, 즉 데이터 스트림(data stream)을 프로세싱하는 파서(parser)가 태그를 인식할 수 있도록, 태그로 사용하기 위해서 규정되고 할당되는 문자 및 숫자 등의 특수 문자가 포함된 문자열(character string)로 구성된다.

XML이 인기 있는 이유 중의 일부는, 그의 확장 가능하고 유동적(flexible)인 신택스(syntax)에 의해서, 문서 개발자들이 태그를 생성하여 명시적 내포형 트리 문서 구조(nested tree document structure)(이 구조는 특정 문서 내의 태그들 사이의 관계에 의해서 결정됨)를 전달할 수 있게 하기 때문이다. 문서 개발자는 애플리케이션 특정 시맨틱(semantics)을 가질 수 있는 자신의 태그를 규정할 수 있다. 이 확장가능성 때문에, 사실상 무한정한 개수의 컨텍스트(contexts)에서 이용되는 여러 서로 다른 타입의 정보를 기술하는 데에 XML 문서를 이용할 수 있다. 다수의 파싱 표기법이 규정되어 왔고, 특정 용도를 위해서 계속적으로 규정될 것이다. 'VoiceXML'은 이러한 파싱물의 일례이다. 본 명세서에서 XML로 참조한 것은, XML 파싱을 및 XML이 파싱된 SGML(Standard Generalized Markup Language)의 파싱을 통의 의미적으로 유사한 표기법을 포함하기 위한 것이다. SGML에 관한 더 많은 정보를 위해서는, ISO 8879: Standard Generalized Markup Language(SGML), 1976를 참조한다. XML에 대한 더 많은 정보를 위해서는, 월드 와이드 웹 상의 <http://www.w3.org/TR/1998/REC-xml-19980210>에서 입수할 수 있는 'Extensible Markup

Language(XML), W3C Recommendation 10-February-1999'를 참조한다.

확장 가능 태그 스펙스는 XML 문서가 예를 들면, 연관된 데이터 값 및 데이터 소스 사이의 전체적 관계의 시맨틱 의미를 전달하는 등을 위해서, 인간이 관독하기에 용이하게 한다. 이 인간 친화적이고, 잘 구성된 포맷은 인간이 임의의 XML 문서를 빠르게 훑어보고 데이터와 그 의미를 이해할 수 있게 한다. 그러나, 대부분의 XML 문서의 원 콘텐츠(raw content)는 인간이 실제로 관찰할 수 있는 대신에, 최종 사용자는, 전형적으로 태그를 제거하고 내장된 데이터 콘텐츠를 디스플레이하는 렌더링 애플리케이션(rendering application)(브라우저 내의 XML 파서(parser) 등)를 이용하여 생성된 것을 관찰하게 된다. 인간 친화적 태그 스펙스의 추가된 오버헤드(overhead)는, 예를 들면 파싱 등의 문서의 프로세싱이 프로세서에게 부담이 되게 한다. 전형적으로, XML 문서는 XML 파서에 의한 DOM(Document Object Model) 트리 표기법으로서 내부적으로 파싱되거나 저장된다. DOM 트리는 대상을 이용하여, 트리에서의 노드, 노드의 특성, 노드의 값 등을 나타내게 하는 트리 표기법으로 물리적으로 저장된다.

다음으로, (예를 들면, 콘텐츠 렌더러(content renderers) 또는 스타일 시트 프로세서(style sheet processors)에 의해)이 트리 표기법을 동작시키는 것에 의해서 변환, 즉, 동작을 수행한다. 예를 들면, 특정 변환은, DOM 트리의 서브 트리(subtree)를 편집(pruning)하는 것에 의해 문서의 요소를 제거하거나, DOM 트리를 횡단하여 요소 이름이 출현하였다는 것을 확인하고, DOM 트리의 적절한 노드에 새로운 이름을 삽입하는 것에 의해, 문서 내의 구성 요소를 리네임(rename)하는 것을 포함할 수 있다(DOM은 월드 와이드 웹 컨소시엄(World Wide Web Consortium)의 W3C)의 권고에 따라서 Document Object Model(DOM) Level 1 Specification, Version 1.0이라는 제목으로 1998년에 출판되었으며, 웹 상의 <http://www.w3.org/TR/REC-DOM-Level-1/>에서 입수할 수 있을, DOM은 Massachusetts Institute of Technology의 상표명임), 변환의 타입은 전형적으로 목표(target)에 의존한다. 예를 들면, 이러한 변환은 의도된 수신자의 등록된 기호에 따라서, 또는 목표 장치, 예를 들면, 웹 가능형 무선 전화기(Web-enabled wireless telephone) 등의 성능에 따라서 수행할 수 있다. 변환은 매우 프로세서 집약적이고 점차 널리 보급되고 있으므로, 더 넓은 범위의 이종 장치(heterogeneous device)가 공통의 데이터 세트에 액세스하도록 모색함에 따라서 더 부담을 주고 있다.

DOM 트리의 생성을 포함하는 파싱, 및 문서의 변환은, 전형적으로 범용 하드웨어 프로세서(general purpose hardware processor)에 의해서 실행되는 전용 소프트웨어(special purpose software)에 의해서 수행된다. 예를 들면, 이들 단계는, 전형적으로 예를 들면, IBM(International Business Machines Corporation of Armonk, New York, U.S.A.)에 의해 제조 또한/또는 배포되고, 표준 PC의 마이크로프로세서 등과 같은 범용 프로세서에 의해서 실행 가능한, WTP(WebSphere® Transcoding Product) 전용 소프트웨어를 사용하는, 네트워크의 에지에 있는 서버에 의해서 수행된다.

일부 실시예에서, 문서 트리를 조작하여, 본 기술 분야에서 일반적으로 알려진 바와 같은 문서 아레이 모델 구조(document array model structure)를 생성할 수 있다. 일반적으로, 아레이 모델에서, 데이터는 조직화되어, 세트의 값을 중의 하나를 고유하게 식별하는 하나 이상의 값을 공급하는 것에 의해 액세스될 수 있는 순차적인 값의 세트를 나타낸다. 따라서, 인간 친화적인 마크업 언어 태그는 트리 모델보다는 아레이 모델로 대표된다. 아레이 모델은 프로세싱을 단순화하고 촉진시킨다.

추가적으로, XML 문서를 XML과 유사한 대신 자향 언어(machine-oriented language)인 mXML 언어로 변환하거나 나타낼 수 있다. mXML 표기법은 인간 친화적 XML 표기법에 비해서 더 소형이므로, 프로세싱 및 전송에 있어서 성능상의 이득을 제공한다.

예를 들면, XML 문서 인식, 콘텐츠 기반의 스타일 시트 선택, 콘텐츠 기반의 라우팅(routing) 및 다른 통상적인 XML 프로세싱 단계 등의 파싱, 변환 및 다른 조작 단계는, 극도로 프로세서 집약형으로서, 범용 프로세서 및 다른 시스템 리소스에게 부담을 준다. 특히, 이러한 프로세싱 단계는, 범용 프로세서가 범용 프로세서를 필요로 하는 다른 작업을 수행하는 것을 방해하거나 지연시킨다.

#### 본 발명의 장래적 설명

본 발명에 따르면, 제 1 측면에서, 마크업 언어로 인코딩된 문서를 효율적으로 프로세싱하는 방법을 제공한다. 이 방법은, 목표에 전달하도록 의도된 문서를 수신하는 단계와, 전용 프로세서를 이용하여 문서를 프로세싱하는 단계와, 프로세싱된 문서를 목표에 전달하여 범용 프로세서에 의해서 추가적으로 프로세싱하게 하는 단계를 포함한다.

바람직하게는, 프로세싱 단계는 문서를 파싱하는 것을 포함한다.

바람직하게는, 프로세싱 단계는 문서에 대한 변환을 수행하는 것을 포함한다.

바람직하게는, 프로세싱 단계는 문서의 아레이 기반 모델을 생성하는 것을 포함한다.

바람직하게는, 프로세싱 단계는 문서의 트리 기반 모델을 생성하는 것을 포함한다.

바람직하게는, 전용 프로세서는 문서를 파싱하도록 구성되는 집적 회로를 포함한다.

바람직하게는, 전용 프로세서는 문서를 파싱하기 위해서 컴퓨터 판독가능 코드를 실행하는, 주요 범용 프로세서와는 별개의 보조 범용 프로세서를 포함한다.

바람직하게는, 전달 단계는, 프로세싱된 문서를 인쇄 회로 기판(printed circuit board)의 버스(bus)를 통해 애플리케이션 프로세서에 대해 통신하는 것을 포함한다.

바람직하게는, 전달 단계는, 프로세싱된 문서를, 통신 네트워크를 거쳐서 목표에 대하여 통신하는 것을 포함한다.

바람직하게는, 목표는 국부 애플리케이션 프로세스(local application process)이다.

바람직하게는, 목표는 원격 장치(remote device)이다.

제 2 측면에서, 본 발명은 마크업 언어로 인코딩된 문서를 효율적으로 프로세싱하는 시스템을 제공하며, 이 시스템은, 메모리와, 이 메모리에 저장된 컴퓨터 판독가능 코드를 실행하기 위해 이 메모리에 동작 가능하게 접속된 범용 프로세서와, 마크업 언어로 인코딩된 문서를 프로세싱하기 위해 이 메모리에 동작 가능하게 접속된 전용 프로세서를 포함하며, 이 전용 프로세서는 전용 방식(dedicated) 프로세서이다.

바람직하게는, 전용 프로세서는 마신 지향 확장 가능 마크업 언어(machine-oriented extensible markup language :  $\mathbb{M}\mathbb{X}\mathbb{L}$ )로 인코딩된 문서를 파싱하도록 구성된다.

바람직하게는, 전용 프로세서는 마신 지향 확장 가능 마크업 언어( $\mathbb{M}\mathbb{X}\mathbb{L}$ )로 인코딩된 문서를 변환하도록 구성된다.

바람직하게는, 전용 프로세서는 문서를 프로세싱하도록 구성된 집적 회로를 포함한다.

이 시스템은, 범용 프로세서에 작동 가능하게 접속되고, 통신 네트워크를 거쳐서 통신할 수 있는 원격 통신 장치(telecommunications device)와, 메모리에 저장되어, 전용 프로세서를 제어하여, 문서를 프로세싱하고, 프로세싱된 문서를 목표에 대하여 통신하는 범용 프로세서에 의해서 실행될 수 있는 제 1 프로그램을 더 포함하는 것이 바람직하다.

이 시스템은, 메모리 내에 저장되어, 마크업 언어로 인코딩된 문서를 인식하고 전용 프로세서를 응답식으로 제어함으로써, 문서를 프로세싱하는 범용 프로세서에 의해서 실행될 수 있는 제 2 프로그램을 더 포함하는 것이 바람직하다.

바람직하게는, 전용 프로세서는, 문서를 프로세싱하기 위한 컴퓨터 판독가능 코드를 실행하는 보조 범용 프로세서를 포함하는 것이 바람직하다.

바람직하게는, 컴퓨터 판독 가능 코드는 문서를 마신 지향 확장 가능 마크업 언어( $\mathbb{M}\mathbb{X}\mathbb{L}$ )로 프로세싱하도록 구성된다.

이 시스템은, 범용 프로세서에 동작 가능하게 접속되고, 통신 네트워크를 거쳐서 통신할 수 있는 원격 통신 장치와, 메모리에 저장되어, 전용 프로세서를 제어하여, 문서를 프로세싱하고, 프로세싱된 문서를 목표에 대하여 통신하는 범용 프로세서에 의해서 실행될 수 있는 제 1 프로그램을 더 포함하는 것이 바람직하다.

이 시스템은, 메모리 내에 저장되어, 마크업 언어로 인코딩된 문서를 인식하고 전용 프로세서를 응답식으로 제어함으로써, 문서를 프로세싱하는 범용 프로세서에 의해서 실행될 수 있는 제 2 프로그램을 더 포함하는 것이 바람직하다.

본 발명은, 메모리 내에 저장된 컴퓨터 판독 가능 코드를 실행하기 위한 범용 프로세서와, 범용 프로세서에 작동 가능하게 접속되어, 범용 프로세서와 통신하기 위한 전용 프로세서-마크업 언어로 인코딩된 문서를 프로세싱하도록 구성된-를 포함하는 인쇄 회로 기판에서 적절히 구현될 수 있다.

바람직하게는, 전용 프로세서는 문서를 프로세싱하도록 구성된 집적 회로를 포함한다.

바람직하게는, 이 프로세싱은 문서의 파싱 또한/또는 변환을 포함한다.

바람직하게는, 전용 프로세서는 보조 범용 프로세서를 포함한다.

인쇄 회로 기판은, 보조 범용 프로세서에 작동 가능하게 접속된 메모리와, 메모리에 저장되어, 문서를 프로세싱하는 보조 범용 프로세서에 의해서 실행될 수 있는 컴퓨터 판독 가능 코드를 더 포함하는 것이 바람직하다. 그에 따라서, 범용 프로세서가 자유롭게 다른 작업을 수행 할 수 있게 하는  $\mathbb{M}\mathbb{X}\mathbb{L}$  등의 마크업 언어로 인코딩된 문서를 프로세싱하기 위한 전용의, 전용 형식 프로세서, 및 적어도, 마신 언어 특성에 의존하는 것에 의해서, 지금까지 알려진 타입의 인간 친화적 소프트웨어 코드에서 비효율성을 제거하거나 감소시키는 것에 의해 프로세싱 단계의 최적화를 제공할 수 있는 하드웨어 기반의 전용 프로세서를 제공하는 것이 바람직하다.

본 발명은, 마크업 언어로 인코딩된 문서를 효율적으로 프로세싱하는 방법을 제공하는 것이 바람직하고, 이 방법은, 문서를 나타내는 여러 가지 기반의 데이터 모델을, 인쇄 회로 기판의 버스를 통해서 애플리케이션 프로세스에 대하여 통신하는 단계를 포함한다.

바람직하게는, 데이터 모델은  $\mathbb{M}\mathbb{X}\mathbb{L}$ 로 인코딩된 문서를 나타낸다.

바람직하게는, 데이터 모델은  $\mathbb{M}\mathbb{X}\mathbb{L}$ 로 인코딩된 문서를 나타낸다.

본 발명은, 전용 형식(전용) 프로세서를 사용하여, 문서를 효율적으로 프로세싱하는 방법 및 장치를 제공하는 것이 바람직하다. 전용 프로세서는, 예를 들면,  $\mathbb{M}\mathbb{X}\mathbb{L}$  문서에 대한 통상적인 파싱, 변환 및 조작 프로세스를 수행할 수 있는 것이 바람직하다. 개념적으로, 문서를 프로세싱하는 데 전용 프로세서를 이용하는 것은, 범용 프로세서가 다른 작업을 수행하는 것을 자유롭게 하고, 이에 따라 시스템 성능을 증가시킨다. 다시 말해, 전용 프로세서는 시스템 리소스에 대해서 경쟁하지 않는 것이 바람직하다.

일 실시예에서, 전용 프로세서는 예를 들면, 하나 이상의 실리콘 칩에 내장된 집적 회로 등의 전용 하드웨어로 구현된다. 이는 하드웨어 구현에서 전형적인 마신 코드(machine code) 및 그 외의 속도 관련 이점을 이용할 수 있게 하기 때문에 특히 유용하다. 예를 들면, 필요한 경우 먼저  $\mathbb{M}\mathbb{X}\mathbb{L}$  문서를  $\mathbb{M}\mathbb{X}\mathbb{L}$ 로 변환하는 것에 의해서, 전용 프로세서가  $\mathbb{M}\mathbb{X}\mathbb{L}$  문서를 프로세싱하도록 구성하는 것에 의해 성능을 향상시킬 수 있다. 이는 하드웨어 기반의 실시예에서 특히 유용하다. 또한, 전용 프로세서를, 여러 가지 기반의 표기법으로 문서를 표현하도록 구성하는 것은, 예를 들면,  $\mathbb{M}\mathbb{X}\mathbb{L}$  기반의 실시예에서, 성능을 강화시키기 위해서 이용될 수 있다. 하드웨어 구현은, 예를 들면, 범용 프로세서와의 통신에 있어서의 패시형 칩(hardwired

(zip) 등과 같이, 단일 프로세서 컴퓨터 시스템에서 특히 유용하다.

다른 실시예에서, 전용 프로세서는 범용 프로세서와, 마크업 언어로 인코딩된 문서를 프로세싱하는 데 통상적으로 이용되어 온 범용 프로세서에 추가하여 제공되는 적절한 소프트웨어를 포함한다. 예를 들면, 다중 프로세서 컴퓨터 시스템 내에서 수 개의 범용 프로세서 중의 하나를 전용 프로세서로 설계할 수 있다.

다른 실시예에서도, 예를 들면, 의도된 목표에 의해서 수신되기 전에 문서를 수신하고 프로세싱하는 프로세싱 장치에서 전용 프로세서를 원격으로 제공할 수 있다. 전용 프로세서가 네트워크에 접근할 수 있는 장치에서는, 수많은 장치를 지지할 수 있어서 수많은 장치의 프로세싱을 오프로딩(offloading)하므로 특히 유용한 것으로 확인된다. 이와 다르게, 하드웨어 기반의 실시예 또는 소프트웨어 기반의 실시예와 마찬가지로, 전용 프로세서는, 예를 들면 단일 장치에서 범용 프로세서와 동일 위치에 배치하는 등으로, 목표 장치 내에 국부적으로 제공할 수 있다.

본 발명은, 마크업 언어로 인코딩된 문서를 효율적으로 프로세싱하는 방법을 제공하는 것이 바람직하고, 이 방법은, 문서를 나타내는 어레이 기반의 데이터 모델을, 인쇄, 회로, 기판의 배스를 통해서 매들리케미 프로세서에 대하여 통신하는 단계를 포함한다. 본 발명은, 마크업 언어로 인코딩된 문서를 효율적으로 프로세싱하는 방법을 제공하는 것이 더 바람직하고, 이 방법은, 목표 전달하도록 의도된 문서를 수신하는 단계와, 전용 프로세서를 사용하여 문서를 프로세싱하는 단계와, 범용 프로세서에 의해 추가적으로 프로세싱하기 위해 목표에 대하여 프로세싱된 문서를 전달하는 단계를 포함한다.

본 발명의 바람직한 실시예는, 이하에서 첨부된 도면을 참조하여, 오직 예시의 방법으로서만 설명될 것이다.

### 도면의 간략한 설명

도 1은 바람직한 실시예에 따라서 문서를 프로세싱하는 예시적인 로직(logic)을 개략적으로 제시하는 흐름도.

도 2a는 도 1에 따라서 문서를 프로세싱하는 예시적인 로직의 제 1 실시예를 제시하는 흐름도.

도 2b는 도 1에 따라 문서를 프로세싱하는 예시적인 로직의 제 2 실시예를 제시하는 흐름도.

도 3은 본 발명이 실행될 수 있는 네트워크형 컴퓨터 환경에 관한 도면.

도 4는 바람직한 실시예에 따른 컴퓨터 워크스테이션 환경의 블록도.

### 실시예

도 1은 본 발명에 따라 문서를 프로세싱하는 예시적인 로직을 개략적으로 제시하는 흐름도(10)를 제공한다. 본 명세서에서 사용된 바와 같이, 프로세싱은, 파싱과, 예를 들면, 스타일 시트의 적용, 또한/또는 문서/문서 트리로부터 데이터를 추가/수정/삭제, 또한/또는 데이터의 포맷 등과 같은 변환과, XML로 인코딩된 문서 인식, 쿼리 엔진 기반의 쿼리 등을 포함하는 그 외의 통상적인 XML 프로세싱 단계를 지칭한다. 예시적인 로직은, 이하에서 논의되는 바와 같이, 본 발명에 따른 전용 프로세서의 하드웨어 기반의 구현 또는 소프트웨어 기반의 구현에 의해서 이용될 수 있다.

도 1에서 도시된 바와 같이, 이 방법은 단계 11 및 단계 12에서 나타난 바와 같이, XML 문서 등의 목표에 전달하도록 의도된 문서를 수신하는 것으로 개시된다. 본 명세서에서 이용된 바와 같이, 목표는, 목표 장치이거나, 웹 브라우저, 기업간(business-to-business) 원격 프로세스 기업 대 고객(business-to-client) 원격 프로세스, 기업 로직 프로세스, 백엔드(back-end) 서버 프로세스, 예지 서버 프로세스, 웹 서비스 정보 교환 프로세스 등과 같은 목표 매들리케미션 프로세스가 될 수 있다. 다음으로, 단계 14에 나타난 바와 같이, 본 발명에 따른 전용 프로세서를 사용하여 문서를 프로세싱한다. 이는, 지금까지 미러한 프로세싱을 수행하는 데 이용되었던 범용 프로세서에 있어서, 시스템 리소스에 전형적으로 상당한 부담을 주었던 집약적 프로세싱에 대한 부담을 덜어준다. 다시 말해, 문서의 프로세싱은, 통상적으로 미러한 프로세싱을 수행해 온 범용 프로세서를 오프로딩한다. 다음으로, 단계 16에서 나타난 바와 같이, 이러한 프로세싱을 수행해 온 범용 프로세서를 오프로딩한다. 다음으로, 단계 16에서 나타난 바와 같이, 예를 들면, 렌더링, 또 다른 변환, 다른 매들리케미션 프로세스로의 부팅 등을 포함하는 후속 프로세싱(post-processing) 등으로 추가 프로세싱하기 위해서, 프로세싱된 문서를 목표에 전달한다. 이러한 후속 프로세싱은, 본 기술 분야에서 잘 알려진 범용 프로세서에 의해서 수행된다. 이는 목표에 이러한 후속 프로세싱을 수행하는 것이 유용하기 때문이다. 그러나, 가장 집약적인 프로세싱이 전용 프로세서에 게로 사실상 오프로딩된다. 이는 시스템 성능을 매우 강화시킨다. 다음에, 단계 17에 나타난 바와 같이, 이 방법은 종료된다.

따라서, 전용 프로세서는, DOM, DDM, MXML 또는 STRAM 및 스타일 시트의 형태로 XML 문서를 입력으로서 수신한다. 추가적으로, 문서와 연관된 노드 트리(node tree)는 인쇄, 회로 기판의 배스를 통해서 매들리케미션 프로세서에 대해서 통신한다. 이는 전용 프로세서가 (다 미하에 논의된 바와 같이) 하드웨어 기반인 소프트웨어 기반인지 여부, 또는 이하에서 논의된 바와 같이, 전용 프로세서가 국부적으로 위치되었는지 원격으로 위치되었는지에 관계없이 발생된다. 또한 이 통신은 문서가 변환되었는지, 그렇지 않으면 파싱, 미하에 조작되었는지, 또는 그것들의 조합인지 여부에 관계없이 발생된다.

예를 들면, 파싱하여 DOM 트리를 생성하는 등의 문서의 프로세싱에 있어서, 인간 친화적 태그 선택스의 주어진 오버헤드는 범용 프로세서에게 부담을 준다. 이 부담은 기업 대 기업(business-to-business; B2B), 기업 대 소비자(business-to-consumer; B2C) 등의 컴퓨터 프로그램 사이에서 상호 교환하기 위해 포맷된 문서 등과 같이, 컴퓨터 프로그램에 의해서만 문서를 볼 수 있을 때에는 불필요하다.

프로세스 효율을 향상시키는 방법 중의 하나는 인간 친화적인 태그 구조를 포기하는 것이다. 본 명세서의 양수인은, XML의 대체물로서 이용하기 위한 대신 지향의 포기법을 이미 개발하였다. 대신 지향의 포기법은, 임의 구조의 문서에 대한 프로세스 시간을 향상시키고, XML의 확장성 및 유동성을 그대로 유지하면서, 또한 동등한 콘텐츠 및 시맨틱 정보를 전달하면서, 저장 요구 사항 및 데이터 상호 교환의 용이성을 감소시킨다. 이 대신 지향의 포기법을 본 명세서에서는 XML로 지칭한다. 따라서, 바람직한 실시예에서, 전용 프로세서는 XML을 이해하고 해석하도록 구성되고, 그에 의해서 프로세스 효율을 얻는다.

DOM 트리의 생성은 프로세스 시간 및 메모리 요구 사항의 관점에서, 수치적으로 값비싸다. 이 트리 지향의 DOM 포기법을 내부 저장 포맷으로서 사용하는 것은 필요한 대상을 저장하는 데 상당한 양의 메모리 또한 또는 저장 공간을 필요로 한다. 추가적으로, 메모리 할당 및 대상 생성, 대상 삭제 및 메모리 해제(de-allocate), 및 그에 대한 동작을 수행하기 위한 트리 구조의 결단을 위해서는, 다수의 컴퓨터 프로그램 명령어가 실행되어야 한다. 이들 명령어를 실행하는 것은, garbage collection(가비지 콜렉션)(이 것으로, 대상이 국부적으로 삭제되거나 해제된 후에, 대상에 의해서 이용되는 공간을 재생할 수 있음)을 수행하도록 주기적으로 실행되는 동작 시스템 호출 명령어(operating system-invoked instructions)가 이행하는 바와 같이, 구성된 문서에서 필요로 하는 프로세스 시간을 증가시킨다.

프로세스 효율을 향상시키는 또 다른 방법은 여러 기반의 포기법을 사용하는 것이다. Apache Software Foundation에 의한 Xalan(XSLT(Extensible Language Transformations) 프로세서는 DOM 트리의 인-메모리 DOM(Document Table Model) 포기법을 제공하는 것에 의해서, DOM 프로세서에서 이용되는 대상의 개수들 어느 정도 감소시킨다. DOM 트리 자체를 저장하기 위해서, 실제 대상(real objects)의 세트 대신에, 메타데이터를 이용한다. 그러나, 문서의 XML 데이터 콘텐츠를 나타내는 데에는 수많은 대상(노드, 노드 값, 특성, 특성 값 등에 대한 대상을 포함함)이 여전히 존재한다. 여러 기반의 프로세서는, 예를 들면, 변환 용도 등을 위해서 트리 구조를 결단하는 것을 용이하게 한다. 따라서, 여러 기반의 프로세서를 전용 프로세서 내에 구현하는 것에 의해서, 추가적인 성능 이득이 실현된다. 매우 바람직한 실시예에서, 전용 프로세서는 여러 기반의 포기법을 사용하여 문서를 프로세스하도록 구성된다.

도 2a는 도 1에 따라서 문서를 프로세스하는 예시적인 로직의 제 1 실시예를 제시하는 흐름도(20)를 제공한다. 도 2a의 예에서, 하드웨어 기반의 전용 프로세서는, 예를 들면, 네트워크 접근 가능 프로세스 장치 내의 전용 칩 또는 칩셋(chipset)로서 원격으로 제공된다. 특히, 전용 프로세서는, 이러한 문서의 소프트웨어 기반의 프로세스를 통상적으로 수행하는 범용 프로세서가 존재하는 장치와는 서로 다른 장치에 제공된다. 예를 들면, 이 장치는, 다수의 장치의 프로세스를 오프로딩하고, 그로 인해 다수의 장치를 지지하기 위한 전용 프로세서를 갖는 네트워크 접근 가능성을 제공하는 것 등에 의해서, 네트워크 기반의 애플리케이션에서 유용하다. 이와 다르게, 즉, 범용 프로세서가 이러한 문서에 대한 소프트웨어 기반의 프로세스를 통상적으로 수행하는 동일 장치 내에서, 전용 프로세서를 국부적으로 제공할 수 있다. 예를 들면, 전용 프로세서는, 연관된 범용 프로세서로부터 프로세스를 오프로딩하기 위해서 국부적으로 제공할 수 있다. 다시 말해, 국부적으로 제공할 때, 전용 프로세서는 동일 장치 내의 범용 프로세서의 프로세스를 오프로딩한다. 원격으로 제공할 때, 전용 프로세서는 원격 장치 내의 범용 프로세서로부터 프로세스를 오프로딩한다. 오프로딩된 프로세스는 사용자에게 투명한 방식으로 수행되는 것이 유용하다.

도 3은 본 발명에 따라 원격으로 제공된 하드웨어 기반의 전용 프로세서가 실시될 수 있는 네트워크형 컴퓨터 환경에 대한 도면이다. 도 3의 네트워크는 이해에서 더 상세히 논의될 것이다. 도 2a의 예에서, 도 3의 게이트웨이 서버(gateway server)(346)는 위에서 설명된 바와 같은 하드웨어 기반의 전용 프로세서를 갖는 프로세스 장치인 것으로 가정한다. 이 예에서, 장치(310a)는 통신 네트워크에 의해서 서버(346)에 접속된 개인용 컴퓨터 장치(310a)이다. 장치(310a)는 데이터 서버(348)에 의해서 서비스되는 XML 문서에 대한 목표 장치인 것으로 가정한다. 더 세부적으로, 장치(310a) 내의 범용 프로세서에 의해서 실행되는 웹 브라우저 소프트웨어는 목표 애플리케이션 프로세스(target application process)인 것으로 가정한다. 전형적인 웹 브라우저 소프트웨어는 XML이 아닌 HTML을 프로세스할 수 있다. 따라서, JAVA 또는 다른 플러그인(plugin) 소프트웨어 애플리케이션은, 전형적으로 장치 내의 범용 프로세서에 의해서 실행되며, 예를 들면, 웹 브라우저 및 범용 프로세서에 의해 해석 및 디스플레이 등의 후속 프로세스에 있어서 XML을 HTML로 변환한다. 이는 목표 장치의 범용 프로세서에게 XML을 HTML로 변환하게 하는 부담을 준다. 따라서, 이 예에서, 서버(346)에는, XML 문서를 프로세스하는 하드웨어 기반의 전용 프로세서를 제공한다. 도 2a의 예 및 도 3에 도시된 바와 같이, 데이터 서버(348)로부터 장치(310a)로 전달될 수 있는 XML 문서를, 도 2a의 단계 22에서 도시된 바와 같이, 중간 프로세스 장치(intermediate processing device)(서버(346))에서 먼저 수신한다(그리고, 하드웨어 또는 소프트웨어 기반의 인식 엔진(recognition engine) 등에 의해 암시적으로 인식됨). 다음으로, 예를 들면 도 2a의 단계 24에서 도시된 바와 같이, 서버(346)의 하드웨어 기반의 전용 프로세서에 의해 파싱하는 등으로 XML 문서를 프로세스한다. 예를 들면, 이러한 파싱은 XML 문서로 나타낸 문서 트리 데이터 모델을, 예를 들면, DOM(document object model) 포맷으로 생성하게 한다. 이와 다르게, 장치(346)의 전용 프로세서는, 문서를 파싱하여 DAM(document array model) 포맷으로 데이터 모델을 생성하도록 구성된다.

선택적으로, 예를 들어 목표 장치에서 필요로 한다면, 도 2a의 단계 26에 도시된 바와 같이, 문서를 추가적으로 프로세스하여 변환을 수행한다. 예를 들면, 이러한 변환은, 전형적으로 개인 휴대 정보 단말기(personal digital assistant: PDA) 또는 도 3의 웹 가능형 무선 전화기(310c) 등과 같은 휴대형 장치로 전달 가능한 콘텐츠의 포맷을 지정하기 위해서 수행된다. 예를 들면, 이러한 변환은, 현재에 있어서 전형적으로, 예를 들면, 도 3의 게이트웨이 서버(346)에 저장되는 IBM의 WTP(WebSphere® Transcoding Product) 소프트웨어에 의해서 수행된다. 이러한 변환을 수행하기 위해서 전용 프로세서를 이용하는 것은, 시스템 성능(예를 들면, 프로세스 장치(346) 등)의 실질적인 향상을 제공한다. 요구되는 특정한 변환은 전형적으로 예를 들면, 낮은 해상도를 제공하거나 이미지를 제공하지 않는 등의 장치 지정(device specific)이거나, 예를 들면 사용자 기호 프로파일에 따라서, 특정한 타입의 콘텐츠를 삭제하는 등의 사용자 지정(user-specific)이다.

도 2a를 점차 참조하면, 파싱 또한/또는 변환 등으로 프로세스된 XML 문서는, 단계 28에 도시된 바와 같이 목표 장치의 범용 프로세서로 후속 프로세스하기 위해서, 통신 네트워크를 거쳐서 목표 장치로 송신된다.

다. 예를 들면, 도 3의 개인용 컴퓨터 장치의 CPU에 의해서, 예를 들면, 웹 브라우저 소프트웨어에 의해서 문서를 디스플레이하기 위해서 이 단계를 수행할 수 있다. 다음으로, 단계 2에서 나타낸 바와 같이, 프로세서는 종료된다. 이러한 방식으로, 통상적으로 문서의 편집/또는 변환과 관련된 목표 장치의 범용 프로세서에 대한 부담은, 예를 들면 서버(346) 등의 프로세서 장치의 전용 프로세서에 이러한 부담을 오프로딩하는 것에 의해서 제거될 수 있다.

도 2b는 도 1b 따라서 문서를 프로세서하는 예시적인 로직의 제 2 실시예를 제시하는 흐름도(30)를 제공한다. 도 2b의 예에서, 소프트웨어 기반의 전용 프로세서를 제공한다. 소프트웨어 기반의 프로세서를 원격으로 제공하는 것이 가능하기는 하지만, 이 예에서, 소프트웨어 기반의 프로세서는, 예를 들면, 동일 장치 내에서 범용 프로세서로부터 프로세싱을 오프로딩하기 위해서, 국부적으로 제공된다. 예를 들면, 이 장치는 통신 네트워크에 의해서 통신할 수 없는 다중 프로세서 시스템 및 시스템에 있어서 유용하다.

또한, 도 3의 네트워크형 컴퓨터 환경은, 도 2b에 제시된 로직에 따라 본 발명을 실시하는 데에 이용될 수 있다. 도 2b의 예에서, 도 3의 서버(346)는 다중 프로세서 프로세싱 장치이고, 소프트웨어 기반의 전용 프로세서는 범용 프로세서 중의 하나를, 서버(346)의 메모리 내에 저장된 소프트웨어를 실행시키는 것에 의해서, XML 프로세싱 작업 전용으로 하는 것에 의해 서버(346) 내에서 구현될 수 있다고 가정한다. 이 예에서는, 위에서 논의된 바와 같이, 워크스테이션(310a)이 목표이고, 프로세싱은 워크스테이션(310a)으로부터 서버(346)의 전용 프로세서(예를 들면, 원격으로 제공되는 전용 프로세서 등)으로 오프로딩하는 도 2b의 예와는 다르게, 서버(346)가 목표인 것으로 가정한다. 이 예에서, 프로세싱은 서버(346)의 범용 프로세서로부터 서버(346)의 전용 프로세서로 오프로딩된다. 본 발명에 따른 프로세싱 장치는 도 4를 참조하여 이하에 상세히 논의될 것이다.

도 2b 및 도 3을 참조하면, 프로세싱은, 도 2b의 단계 31, 단계 32에서 나타낸 바와 같이, 프로세싱 장치(이 경우의 목표인)에서 XML 문서를 수신하는 것으로 제시된다. 다음으로, XML 문서를 도 2b의 단계 34 및 단계 36에서 나타낸 바와 같이, 전용 프로세서로 파싱하고 변환한다. 이들 단계는 도 2a의 단계 24 및 단계 26과 유사하다. 그러나, 이 예에서, 이들 단계는 도 4의 프로세싱 장치의 국부 전용 프로세서(432)에 의해서, 워크스테이션(410)의 메모리(418, 430) 내에 저장된 소프트웨어를 실행하는 범용 프로세서에 의해서 수행된다. 다음으로, 편집/또는 변환된 XML 문서는, 단계 38에서 나타낸 바와 같이, 예를 들면, 후속 프로세싱을 위해서 범용 프로세서쪽으로 전달된다. 예를 들면, 이 단계는, 문서를 나타내는 노드 트리들, 인쇄 흐름 기판의 배치를 통해서 국부적으로 진행되는 애플리케이션 프로세스에 대하여 통신하는 것을 포함한다. 전용 프로세서는 국부적으로 제공되기 때문에, 이 단계는, 도 2a의 예와 같이, 통신 네트워크를 거쳐서 프로세싱된 문서를 송신하는 것을 포함할 필요는 없다.

요약하면, 도 3의 서버(346) 내에서 특수 프로세서를 국부적으로 제공하여, 서버(346) 내의 범용 프로세서로부터 프로세싱을 오프로딩(국부 매립(local embodiment))하거나, 예를 들면 310a 등의 워크스테이션으로부터 프로세싱을 오프로딩(원격 매립(remote embodiment))할 수 있다. 국부 매립은 원격 매립 기간에, 하드웨어 구현(전용 칩 또는 칩셋) 또는 소프트웨어 구현(추가적인 범용 프로세서 및 전용 소프트웨어)의 어느 쪽에도 전용 프로세서를 제공할 수 있다.

#### 네트워크

도 3은, 본 발명이 실시될 수 있는 예시적인 데이터 프로세싱 네트워크(340)를 도시한다. 데이터 프로세싱 네트워크(340)는, 예를 들면, 410a, 410b, 410c 등의 복수의 개별적인 워크스테이션/장치를 각각 포함할 수 있는 무선 네트워크(342) 및 네트워크(344) 등의 복수의 개별 네트워크를 포함할 수 있다. 추가적으로, 당업자라면, 호스트 프로세서에 결합된 복수의 지능형 워크스테이션(intelligent workstations)을 포함할 수 있는 하나 이상의 LAN(도시하지 않음)이 포함될 수 있다는 것을 인식할 것이다.

또한, 네트워크(342, 344)는, 게이트웨이 컴퓨터(346), 또는 애플리케이션 서버(application server)(347)(데이터 리포지터리(data repository)(348)에 액세스할 수 있음), 등의 메인프레임(main frame) 컴퓨터 또는 서버를 포함할 수 있다. 게이트웨이 컴퓨터(346)는 각 네트워크(344)로의 입력 지점(point of entry)으로서의 기능을 한다. 게이트웨이 컴퓨터(346)는 통신 링크(350a)에 의해서 다른 네트워크(342)로 결합되는 것이 바람직할 수 있다. 또한, 게이트웨이 컴퓨터(346)는, 통신 링크(350b, 350c)를 이용하여 하나 이상의 워크스테이션(예를 들면, 310d, 310e)에 직접적으로 결합될 수 있다. 게이트웨이 컴퓨터(346)는 IBM의 네트워크 프로세서 등의 임의의 적절한 프로세서를 이용하여 구현될 수 있다. 예를 들면, 게이트웨이 컴퓨터(346)는, IBM p 시리즈(P/6000) 또는 x 시리즈(Netfinity) 컴퓨터 시스템과, IBM으로부터 입수가능한 Enterprise Systems Architecture/390, Enterprise Systems Architecture/390 컴퓨터 등을 이용하여 구현될 수 있다. 애플리케이션에 따라서, Application System/400(AS/400으로도 알려져 있음) 등의 미드레인지(midrange) 컴퓨터를 채용할 수 있다. Enterprise Systems Architecture/390은 IBM의 상표명이고, Enterprise Systems Architecture/390, Application System/400 및 AS/400은 IBM의 등록 상표명이다. 이들은 본 바람직한 실시예에서 이용될 수 있는 컴퓨터의 대표적인 타당에 불과하다.

또한, 게이트웨이 컴퓨터(346)는 저장 소자(데이터 리포지터리(348) 등)(349)에 결합될 수 있다. 또한, 게이트웨이(346)는, 하나 이상의 워크스테이션/장치(310d, 310e) 및 애플리케이션 서버(347) 등의 서버에 직접적으로 또는 간접적으로 접속될 수 있다.

당업자라면, 게이트웨이 컴퓨터(346)를 네트워크(342)로부터 지리적으로 매우 먼 거리에 위치시킬 수 있고, 이와 유사하게, 워크스테이션/장치를 네트워크(342, 344)로부터 상당한 거리에 위치시킬 수 있다는 것을 인식할 것이다. 예를 들면, 네트워크(342)를 헬리콥터에 위치시키는 한편, 게이트웨이(346)를 역시에 위치시킬 수 있고, 하나 이상의 워크스테이션/장치(310)를 뉴욕에 위치시킬 수 있다. 워크스테이션/장치(310)는, 셀룰라 폰(cellular phone), 무선 주파수 네트워크(radio frequency networks), 위성 네트워크 등의 다수의 대체 접속 매체를 거쳐서 TCP/IP(Transmission Control Protocol/Internet Protocol) 등의 네트워크 프로토콜을 이용하는 무선 네트워크(342)에 접속될 수 있다. 무선 네트워크(342)는, IP, X.25, 프레임 중계(Frame Relay), ISDN(Integrated Services Digital Network), PSTN(Public Switched



Telephone Network) 등을 거쳐서, TCP 또는 UDP(User Datagram Protocol) 등의 네트워크 접속(350a)을 이용하는 게이트웨이(345)에 접속되는 것이 바람직하다. 이와 다르게, 워크스테이션/장치(310)는 다이얼 접속(350b 또는 350c)을 이용하여, 게이트웨이(345)에 직접적으로 접속될 수 있다. 또한, 무선 네트워크(342) 및 네트워크(344)는, 도 3에 도시된 것과 유사한 방식으로 하나 이상의 네트워크(도시하지 않음)에 접속될 수 있다.

본 바람직한 실시예는, 네트워크 환경에서, 클라이언트 컴퓨터 또는 서버에서 이용되거나 (예를 들면, 파일을 작성하거나, 책상 가상 저장 매체 등에 의해서, 네트워크 접속을 거쳐서 수신된 파일을 프로세싱하기 위한) 독립형 워크스테이션(standalone workstation)에 이용될 수 있다(본 명세서에서 참조한 클라이언트 및 서버 장치는 설명을 목적으로 한 것으로서, 한정되는 것이 아님을 주의하라). 본 바람직한 실시예는 다른 네트워크 모델에도 유용하게 이용될 수 있을 것이다. 네트워크 환경에서 이용될 때, 클라이언트 및 서버 장치는, 유선(wireline) 접속 또는 무선(wireless) 접속을 이용하여 접속될 수 있다. 유선 접속은 케이블 및 전화선 등의 물리적 매체를 이용하는 것인 반면, 무선 접속은 위성 링크(satellite links), 고주파 및 적외선 파장 등의 매체를 이용한다. 여러 접속 기법은, 전화선을 거쳐 접속을 설정하기 위해 컴퓨터 모뎀을 이용하는 것, 토큰 링(Token Ring) 또는 이더넷(Ethernet) 등의 LAN 카드를 이용하는 것, 무선 접속을 설정하기 위해 셀룰러 모뎀(cellular modem)을 이용하는 것과 같이 이들 여러 매체를 가지고 이용할 수 있다. 워크스테이션 또는 클라이언트 컴퓨터는, 프로세싱(또한 선택적으로, 통신) 기능을 갖는, 랩탑(laptop), 휴대형 또는 모바일 컴퓨터, 차량 탑재형 장치, 데스크톱 컴퓨터, 메인프레임 컴퓨터 등을 포함하는 임의 타입의 컴퓨터 프로세싱될 수 있다. 이와 유사하게, 서버는 프로세싱 및 통신 기능을 갖는 임의의 개수의 서로 다른 타입의 컴퓨터 중의 하나일 수 있다. 이들 기법은 본 기술 분야에서 잘 알려져 있고, 그의 용도를 용이하게 이용할 수 있게 하는 하드웨어 장치 및 소프트웨어를 용이하게 입수할 수 있다.

#### 프로세싱 장치

도 4는 본 바람직한 실시예에 따른 프로세싱 장치(410)의 블록도이다. 예시적인 프로세싱 장치(410)는 앞서 논의된 바와 같이, 도 3의 워크스테이션(310a) 또는 서버(346)를 대표하는 것이다. 이 블록도는 국부 구현 또는 원격 구현을 위한 하드웨어를 나타낸다. 그러나, 예를 들면, 메모리에 저장된 적절한 소프트웨어가 제공되어, 국부 또한/또는 원격 범용 프로세서로부터 프로세싱을 오프로딩하도록 워크스테이션을 구성한다.

본 기술 분야에서 잘 알려진 바와 같이, 도 4의 워크스테이션은, 예를 들면, 관련된 주변 장치를 포함하는 개인용 컴퓨터 등의 단일 사용자 컴퓨터 워크스테이션(410) 등의 대표적인 프로세싱 장치를 포함한다. 워크스테이션(410)은 범용 마이크로프로세서(412); 및 알려진 기법에 따라서 마이크로프로세서(412)와 워크스테이션(410)의 구성 요소 사이를 접속하고 통신을 가능하게 하기 위해 사용되는 버스(414)를 포함한다. 워크스테이션(410)은 전형적으로, 버스(414)를 거쳐서 마이크로프로세서(412)를, 키보드(418), 마우스(420) 또한/또는 접촉 감응성 스크린(touch sensitive screen), 디지털화 입력 패드(digitized entry pad) 등과 같은 임의의 사용자 인터페이스 장치인 그 외의 인터페이스 장치(422) 등의, 하나 이상의 인터페이스 장치로 접속시키는 사용자 인터페이스 어댑터(user interface adapter)(416)를 포함한다. 또한, 버스(414)는 LCD 스크린 또는 모니터 등과 같은 디스플레이 장치(424)를 디스플레이 어댑터(426)를 거쳐서 마이크로프로세서(412)에 접속시킨다. 또한, 버스(414)는 마이크로프로세서(412)를, 하드 드라이브, 디스켓 드라이브, 테이프 드라이브 등을 포함할 수 있는 메모리(428) 및 장기 기억 장치(long-term storage)(430)(통칭하면, '메모리')에 접속시킨다.

워크스테이션(410)은, 예를 들면 통신 채널 또는 모뎀(434) 등을 거쳐서 다른 컴퓨터 또는 컴퓨터의 네트워크와 통신할 수 있다. 이와 다르게, 워크스테이션(410)은, 434에서 CCPD(cellular digital packet data) 카드 등의 무선 인터페이스를 이용하여 통신할 수 있다. 워크스테이션(410)은 LAN 내의 그러한 컴퓨터 또는 WAN(wide area network)과 연결될 수 있고, 또는 워크스테이션(410)은 다른 컴퓨터가 존재하는 클라이언트/서버 장치 내에서의 클라이언트 동일 수 있다. 적절한 통신 하드웨어 및 소프트웨어뿐만 아니라, 이들 모든 구성은 본 기술 분야에서 알려져 있다.

본 바람직한 실시예에 따르면, 전용 프로세서(432)는, 범용 마이크로프로세서(412), 메모리(428), 장기간 저장 장치(430) 등과 버스(414)에 의한 통신에 제공된다. 국부 범용 프로세서로부터 프로세싱을 오프로딩하기 위해서 워크스테이션(410)을 사용할 때, 워크스테이션(410)은, 범용 프로세서에 비해, 전용 프로세서가 근접성 또한/또는 우선 순위를 가지고 있기 때문에, 프로세싱 작업이 오프로딩되는 것에 의해, 특별한 성능 향상을 제공한다.

도 2b의 소프트웨어 기반의 예에서, 전용 프로세서(432)는 메모리(428) 또한/또는 저장 장치(430) 내에 저장된 프로세싱 소프트웨어를 실행하는 전용의 범용 마이크로프로세서를 포함한다. 하드웨어 기반의 실시예에서, 전용 프로세서(432)는 전용 칩 또는 칩셋을 포함한다. 어느 실시예에서도, 전용 프로세서를 여러이 기반의 프로세싱 또한/또는 예를 들면, XML 등의 마신 언어 기반의 프로세싱에 이용하도록 구성하는 것에 의해서 추가적인 성능 이득을 실현할 수 있다. 추가적인 성능 이득은, 하드웨어 기반의 실시예를 이러한 여러이 기반의 프로세싱 또한/또는 XML에 이용하도록 최적화하는 것에 의해서 실현될 수 있다. 예를 들면, 전용 프로세서(432)는, 전용 하드웨어와, 예를 들면, 스태일 시트 업데이트를 프로세싱하는 것, 개인화(personalization) 또는 커맨드/데이터를 처리하는 것, 캐싱(caching) 등의 부가적인 소프트웨어 기능을 취급하기 위해서, 비반복적인 작업을 전용 프로세서로부터 오프로딩하는 범용 프로세서를 더 포함할 수 있는 마이크로코드(microcode)와의 조합을 통해서 구현될 수 있다.

#### (5) 공구의 범위

##### 원구현 1

마크업 언어(markup language)로 인코딩된 문서를 효율적으로 프로세싱하는 방법에 있어서,

목표(target)에 전달되도록 의도된 문서를 수신하는 단계와,

전용 프로세서(special purpose processor)를 이용하여 상기 문서를 프로세싱하는 단계와,

상기 프로세싱된 문서를 상기 목표에 전달하여 범용 프로세서(general purpose processor)에 의해서 추가적으로 프로세싱하게 하는 단계

를 포함하는 마크업 언어로 인코딩된 문서의 효율적인 프로세싱 방법.

## 청구항 2

제 1 항에 있어서,

상기 프로세싱 단계는 상기 문서를 파싱(parsing)하는 것을 포함하거나, 상기 문서에 대한 변환을 수행하는 것을 포함하는 마크업 언어로 인코딩된 문서의 효율적인 프로세싱 방법.

## 청구항 3

제 1 항 또는 제 2 항에 있어서,

상기 프로세싱 단계는, 상기 문서의 머레이 기반 모델(array-based model)을 생성하는 것을 포함하거나, 상기 문서의 트리 기반 모델(tree-based model)을 생성하는 것을 포함하는 마크업 언어로 인코딩된 문서의 효율적인 프로세싱 방법.

## 청구항 4

제 1 항 내지 제 3 항 중 어느 한 항에 있어서,

상기 전용 프로세서는 상기 문서를 파싱하도록 구성된 집적 회로를 포함하거나, 상기 문서를 파싱하기 위한 컴퓨터 판독 가능 코드(computer readable code)를 실행하는 보조 범용 프로세서를 포함하고,

상기 보조 범용 프로세서는 주요 범용 프로세서와는 구별되는

마크업 언어로 인코딩된 문서의 효율적인 프로세싱 방법.

## 청구항 5

제 4 항에 있어서,

상기 전용 단계는, 프로세싱된 상기 문서를 인쇄 회로 기판(printed circuit board)의 버스(bus)를 통해서 매들리케이션 프로세서에 대하여 통신하는 것을 포함하거나, 프로세싱된 상기 문서를, 통신 네트워크를 거쳐서 목표에 대하여 통신하는 것을 포함하는 마크업 언어로 인코딩된 문서의 효율적인 프로세싱 방법.

## 청구항 6

마크업 언어(markup language)로 인코딩된 문서를 효율적으로 프로세싱하는 시스템에 있어서,

하도리외,

상기 메모리에 동작 가능하게 접속되어, 상기 메모리에 저장된 컴퓨터 판독가능 코드를 실행하는 범용 프로세서와,

상기 메모리에 동작 가능하게 접속되어, 상기 마크업 언어로 인코딩된 문서를 프로세싱하는 전용 프로세서

를 포함하며,

상기 전용 프로세서는 전용 방식(dedicated) 프로세서인

마크업 언어로 인코딩된 문서의 효율적인 프로세싱 시스템.

## 청구항 7

제 6 항에 있어서,

상기 전용 프로세서는, 머신 지향 확장 가능 마크업 언어(machine-oriented extensible markup language : mxml)로 인코딩된 문서를 파싱하도록 구성되거나, 머신 지향 확장 가능 마크업 언어(mxml)로 인코딩된 문서를 변환하도록 구성되는 마크업 언어로 인코딩된 문서의 효율적인 프로세싱 시스템.

## 청구항 8

제 6 항 또는 제 7 항에 있어서,

상기 전용 프로세서는 상기 문서를 프로세싱하도록 구성된 집적 회로를 포함하는 마크업 언어로 인코딩된 문서의 효율적인 프로세싱 시스템.

**청구항 9**

제 6 항 내지 제 8 항 중 어느 한 항에 있어서,

상기 범용 프로세서에 작동 가능하게 접속되고, 통신 네트워크를 거쳐서 통신할 수 있는 원격 통신 장치 (telecommunications device)와,

상기 범용 프로세서에 의해서 실행 가능하게 상기 메모리 내에 저장되며, 상기 전용 프로세서를 제어하여 상기 문서를 프로세싱하고 프로세싱된 상기 문서를 목표에 대하여 통신하는 제 1 프로그램  
을 더 포함하는 마크업 언어로 인코딩된 문서의 효율적인 프로세싱 시스템.

**청구항 10**

제 9 항에 있어서,

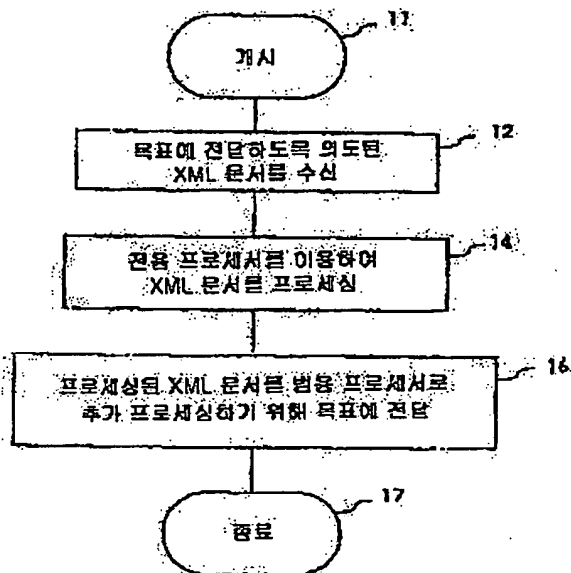
상기 범용 프로세서에 의해서 실행 가능하게 상기 메모리 내에 저장되며, 마크업 언어로 인코딩된 상기 문서를 인식하고 상기 전용 프로세서를 응답식으로 제어함으로써 상기 문서를 프로세싱하는 제 2 프로그램

을 더 포함하는 마크업 언어로 인코딩된 문서의 효율적인 프로세싱 시스템.

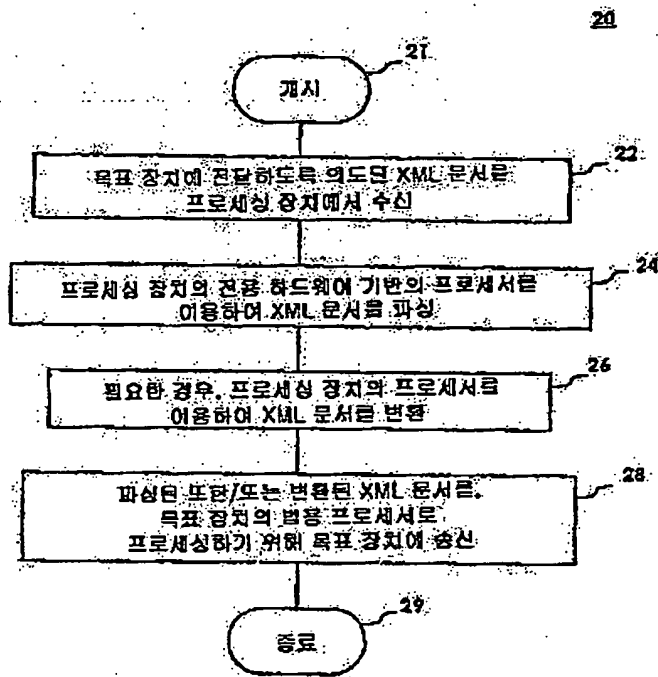
**도면**

**도면**

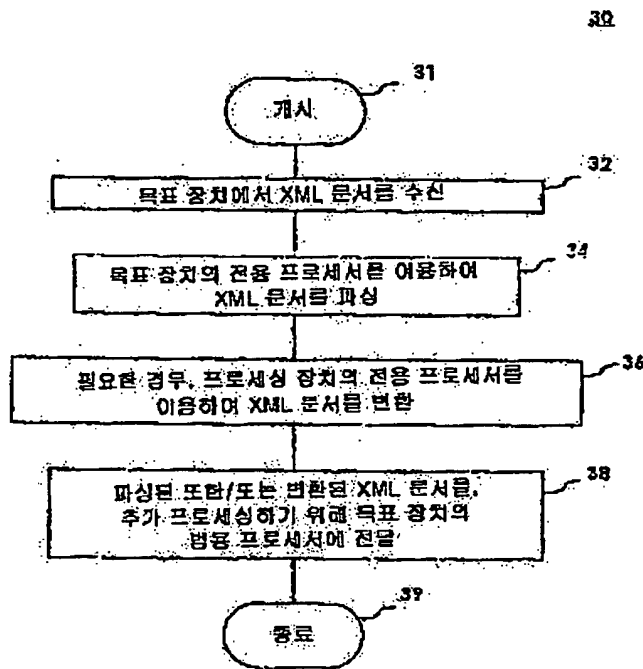
10



도 20



도 21



도 4

